

|  |
| --- |
| Sengès Camille – Quinaou Eddy – Sanchez Stéphane |
| Conception projet Chat CORBA |
| M2 STRI 2014 |

|  |
| --- |
|  |

Mise en place de l'application

L'infrastructure mise en place au sein de notre messagerie sera construite autour d'un système hiérarchique.

En effet, il existe 3 niveaux au sein de cette architecture.

- le niveau 1 : correspond au niveau le plus élevé en termes d'autorité. Il s'agit du niveau regroupant les éléments racines qui sont automatiquement "de confiance".

- le niveau 2 (intermédiaire) : correspond, comme son nom l'indique, au niveau intermédiaire entre les éléments racines et les éléments de niveau 3 connectés aux utilisateurs.

- le niveau 3 : regroupe le trinôme d'éléments connectés directement à l'utilisateur de la messagerie.

Fonctionnement du trinôme d'éléments :

- **AC : Autorité de Certification.** Il regroupe les fonctionnalités de création de certificat, publication des certificats dans les listes de certificats (LCR, LC), révocation et suspension de certificats ainsi que la vérification de certificats de niveaux supérieur.

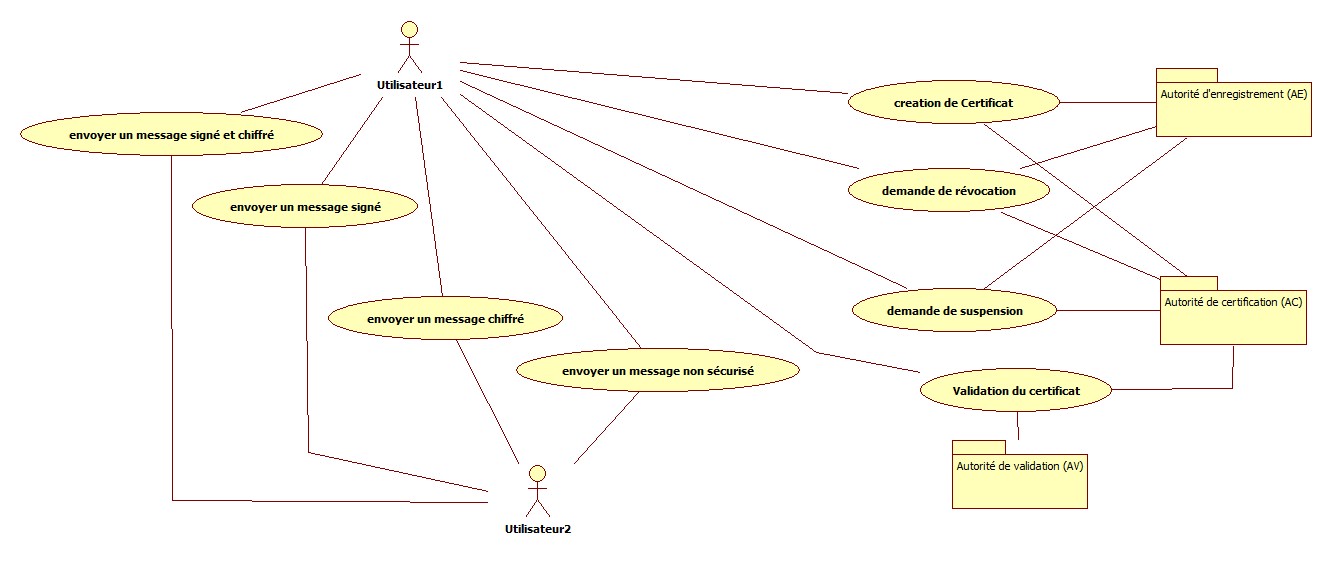
- **AE : Autorité d'Enregistrement**. Cet élément regroupe les fonctionnalités d'enregistrement de certificat, transmission des demandes de révocation et de suspension ainsi que la vérification d'identité d'un utilisateur.

- **AV : Autorité de Validation.** L'AV est l'élément qui réalise les validations de certificats demandées par les utilisateurs.

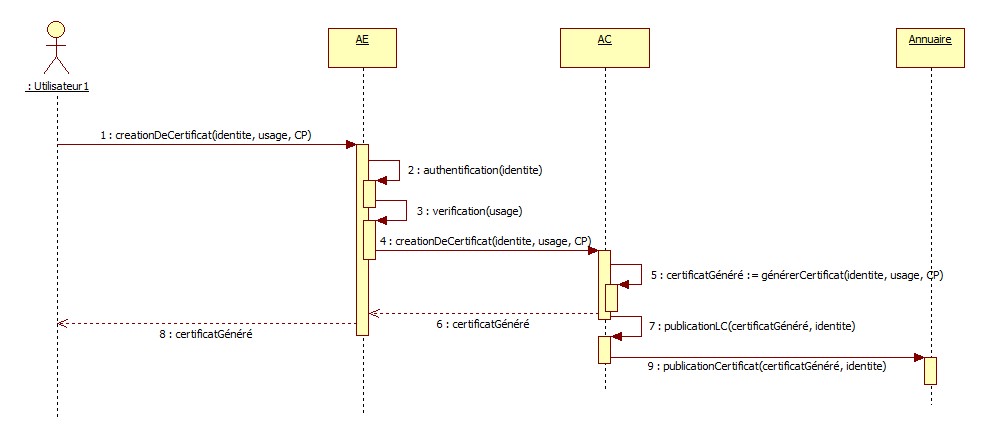
Il faut prendre en compte que pour un AC, nous avons un AE et un AV qui sont liés également.

Diagramme UML

**- Diagramme de cas d'utilisation :**



**- Diagramme de séquence détaillé : "Création de certificat"**



**Description :**

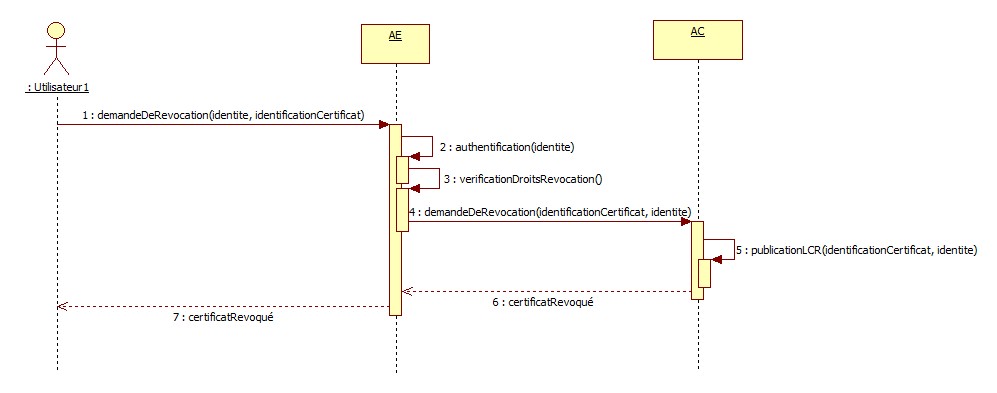
Ce diagramme permet de comprendre les différentes étapes nécessaires pour l'utilisateur lorsqu'il souhaite créer un certificat.

Pour cela, il va dans un premier temps contacter son AE référant situé au-dessus de lui. L'AE va dans un second temps effectuer un ensemble de vérification locale et contacter dans un deuxième temps l'AC associé à l'AE.

L'AC pourra ensuite publier ce nouveau certificat au niveau de sa liste de certificat et transmettre ce certificat au niveau de l'annuaire.

**Postulat de départ :**

Nous considérons que le paramètre "usage" regroupe les droits que l'utilisateur souhaite obtenir.

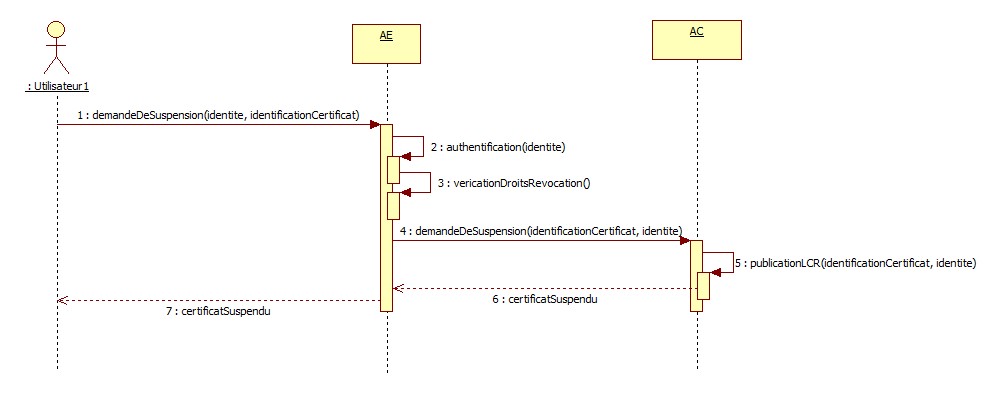
- **Diagramme de séquence détaillé : "Demande de révocation"**

**Description :**

Ce diagramme permet de comprendre l’enchainement des différentes opérations nécessaires à la demande de révocation d'un certificat.

La première étape consiste à l'envoi d'une demande de révocation par l'utilisateur à son AE référant. Après que ce dernier est effectué différentes vérifications localement, la demande est transmise à l'AC de même niveau qui peut ensuite publier ce certificat au sein de la liste des certificats révoqués (LCR).

- **Diagramme de séquence détaillé : "Demande de suspension"**



**Description :**

Ce diagramme permet de comprendre l’enchainement des différentes opérations nécessaires à la demande de suspension d'un certificat.

En effet, dans notre conception, nous avons décidés de découper les demandes de révocation et de suspension en deux étapes distinctes.

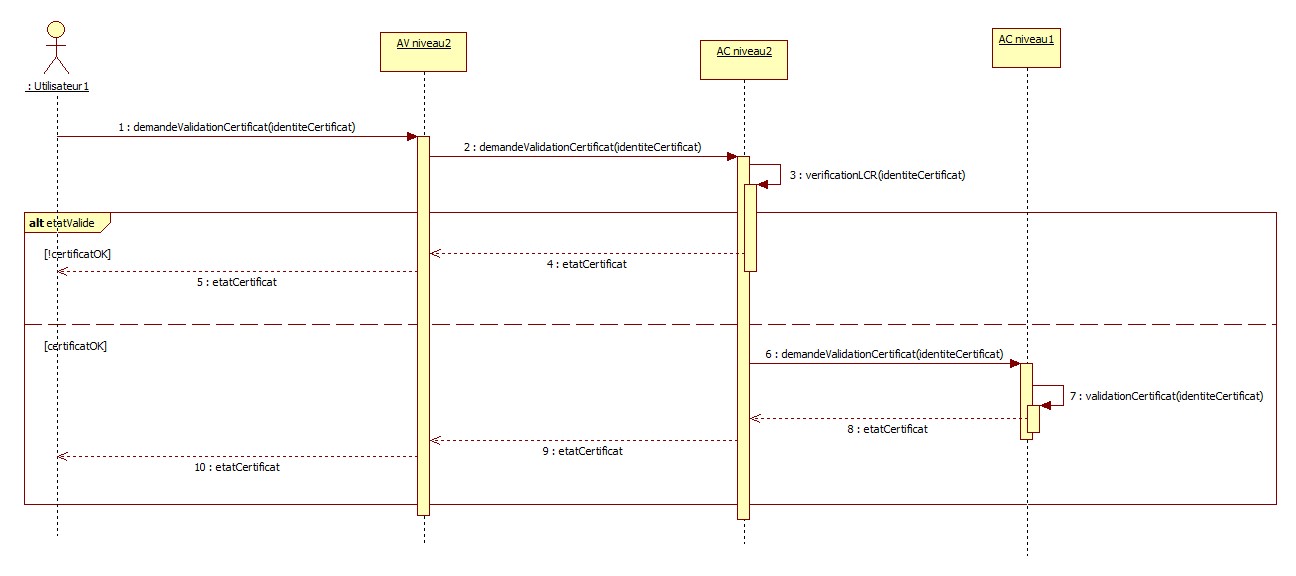
L’enchainement des différentes opérations suit le même déroulement que la demande de révocation.

**Postulat :**

Concernant la publication des certificats suspendus, nous avons décidés de mettre en place un argument au sein de la LCR permettant d’indiquer la nature du certificat : suspendu ou révoqué.

**Postulat pour la révocation et la suspension :**

Lorsqu’un certificat est révoqué ou suspendu, ce dernier est automatiquement retiré de la LC ( Liste des Certificats) et ajouté dans la LCR.

- **Diagramme de séquence détaillé : "Demande de validation de certificat"**

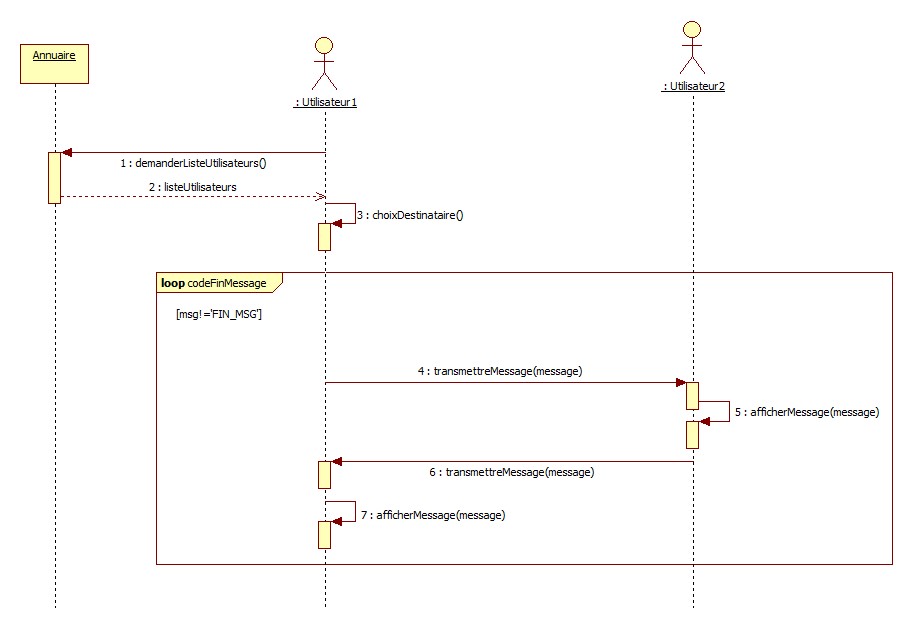
**Description**:

Ce diagramme permet de réaliser la validation d'un certificat suite à la demande d'un utilisateur au cours d'un échange de message sécurisé. L'AV associé à l'utilisateur reçoit la demande et la transmet à son AC. Ce dernier l'envoi ensuite vers son AC référant de niveau inférieur.

A chaque niveau de la hiérarchie, une validation intermédiaire du certificat de l'AC de niveau supérieur est effectuée.

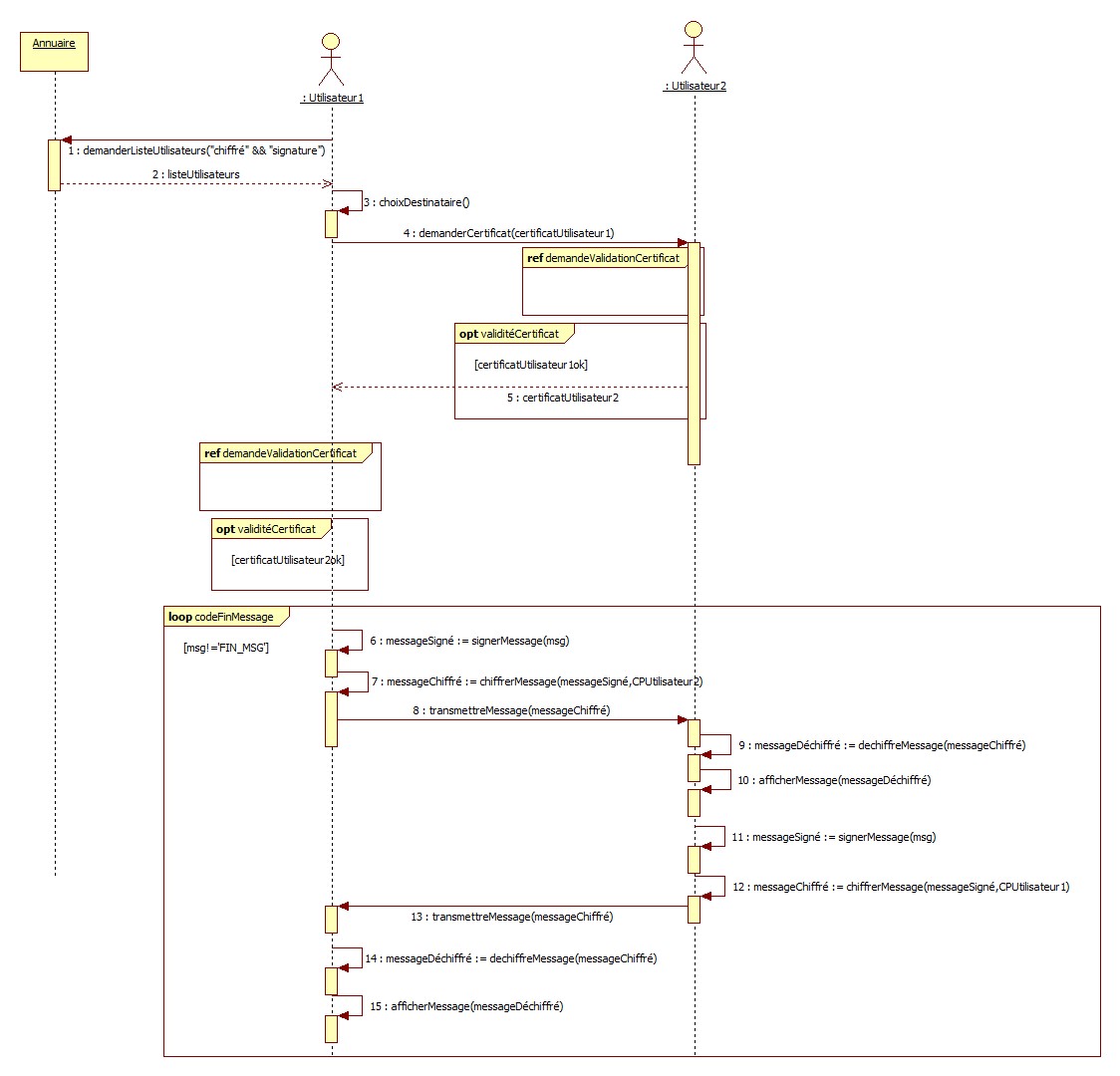
On peut alors remonter tous les niveaux d'AC jusqu'à l'élément racine qui s'auto-validera.

- **Diagramme de séquence détaillé : "Envoyer un message"**



**Description**:

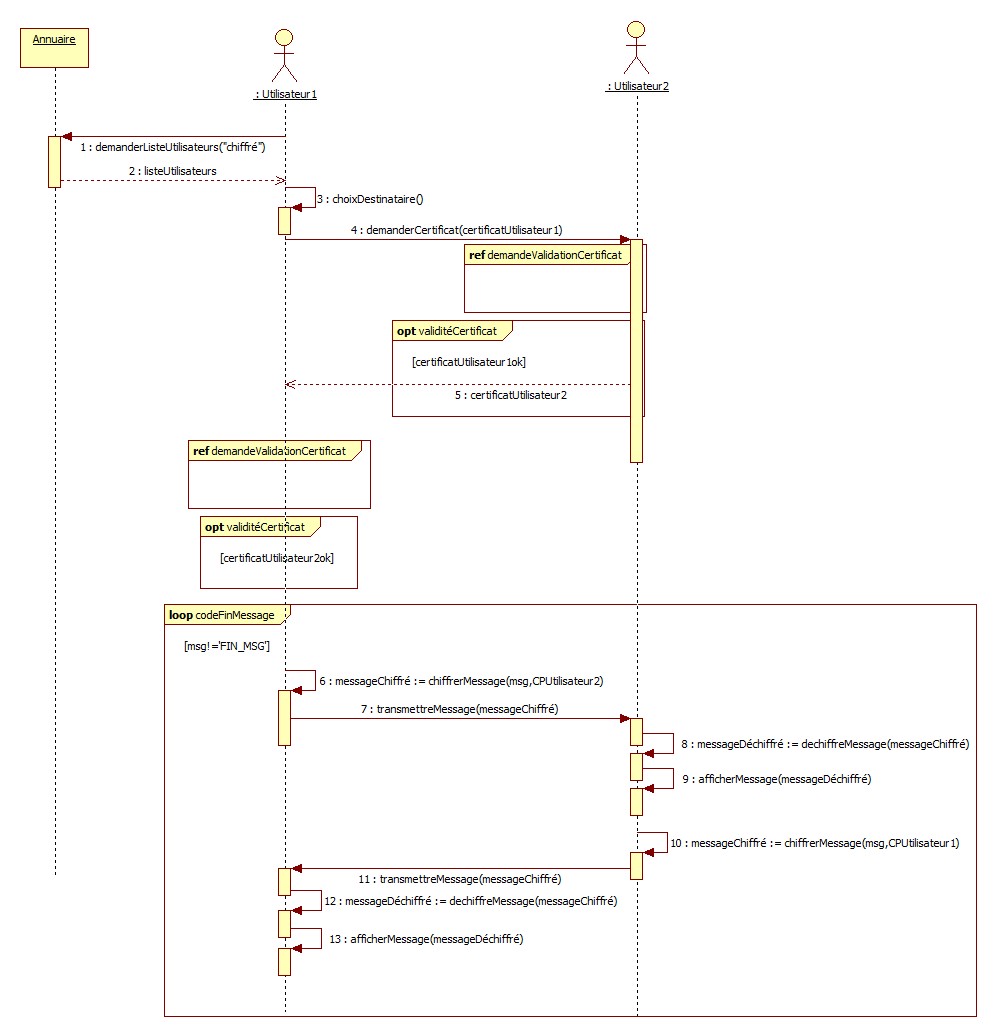
Ce diagramme permet d'envoyer un message normal sans protection à un autre utilisateur. L'utilisateur va dans un premier temps demander à l'annuaire la liste des utilisateurs avec lesquels il va pouvoir démarrer une conversation en mode non sécurisé. Puis tant que l'utilisateur ne marque pas le code de fin "FIN\_MSG" alors celui-ci pour communiquer avec un autre utilisateur.

- **Diagramme de séquence détaillé : "Envoyer message chiffré et signé"**

**Description**:

Ce diagramme permet d'envoyer un message signé et chiffré à un autre utilisateur. L'utilisateur va dans un premier temps demander à l'annuaire la liste des utilisateurs avec lesquels il va pouvoir démarrer une conversation en mode sécurisé (signé et chiffré). L’utilisateur initiateur de la communication va ensuite choisir un destinataire parmi cette liste. Puis une étape de validation des certificats des deux communicants va être effectuée de part et d’autre.

Si la validation s’est bien effectuée, l’initiateur de la conversation effectue la signature de son message avant de le chiffrer avec la clé publique du destinataire. Puis ce dernier pourra alors transmettre ce message sécurisé ». Le destinataire pourra alors déchiffrer ce message à l’aide de sa clé privée et l’afficher ensuite. A la fin de cette étape, le second utilisateur pourra utiliser la même procédure pour répondre sans effectuer les étapes de validation de certificat.

**- Diagramme de séquence détaillé : "Envoyer un message chiffré"**

**Description**:

Ce diagramme permet d'envoyer un message chiffré à un autre utilisateur. L'utilisateur va dans un premier temps demander à l'annuaire la liste des utilisateurs avec lesquels il va pouvoir démarrer une conversation en mode chiffré. L’utilisateur initiateur de la communication va ensuite choisir un destinataire parmi cette liste. Puis une étape de validation des certificats des deux communicants va être effectuée de part et d’autre.

Si la validation s’est bien effectuée, l’initiateur de la conversation effectue le chiffrement de son message avec la clé publique du destinataire. Puis ce dernier pourra alors transmettre ce message « sécurisé ». Le destinataire pourra alors déchiffrer ce message à l’aide de sa clé privée et l’afficher ensuite. A la fin de cette étape, le second utilisateur pourra utiliser la même procédure pour répondre sans effectuer les étapes de validation de certificat.

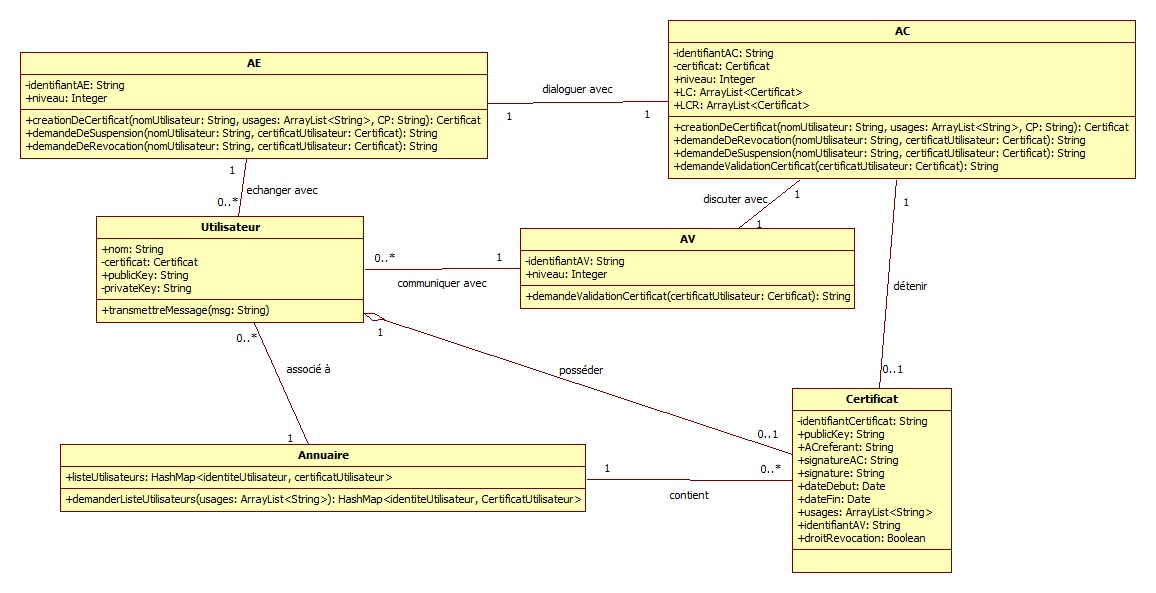
- Diagramme de séquence détaillé : "Envoyer un message signé"

**Description**:

Ce diagramme permet d'envoyer un message signé à un autre utilisateur. L'utilisateur va dans un premier temps demander à l'annuaire la liste des utilisateurs avec lesquels il va pouvoir démarrer une conversation en mode chiffré. L’utilisateur initiateur de la communication va ensuite choisir un destinataire parmi cette liste. Puis une étape de validation des certificats des deux communicants va être effectuée de part et d’autre.

Si la validation s’est bien effectuée, l’initiateur de la conversation effectue la signature de son message. Puis ce dernier pourra alors transmettre ce message « signé ». Le destinataire pourra alors afficher ce message. A la fin de cette étape, le second utilisateur pourra utiliser la même procédure pour répondre sans effectuer les étapes de validation de certificat.

**- Diagramme de classes**



**Contrat IDL :**

Module MessagerieSecurisee

{

**exception** CertificatIntrouvable {

string raisonCertificatIntrouvable;

};

**exception** CertificatDejaCree {

string raisonCertificatDejaCree;

};

**type def struct** Date

{

short jour;

short mois;

short annee;

};

**typedef sequence** <string> ListUsages;

**typedef sequence** <Certificat> ListLC;

**typedef sequence** <Certificat> ListLCR;

**type def struct** Certificat

{

private string identifiantCertificat;

string publicKey;

string ACreferant;

string signatureAC;

string signature;

Date dateDebut;

Date dateFin;

ListUsages usages;

string identifiantAV;

Boolean droitRevocation;

};

**interface** Utilisateur

{

**attribute** string nomUtilisateur;

**private** **attribute** Certificat certificatUtilisateur;

**attribute** string publicKey;

**private** attribute string privateKey;

void transfererMessage (in string message);

Certificat demanderCertificat (in Certificat

certificatDestinataire);

};

**interface** Annuaire

{

**attribute** HashMap<identiteUtilisateur,

certificatUtilisateur> listUtilisateurs;

HashMap<identiteUtilisateur,

certificatUtilisateur> demanderListeUtilisateurs(in

ListUsages usages);

};

**interface** AV

{

**private** **attribute** string identifiantAV;

**attribute** short niveau;

string demandeValidationCertificat(in Certificat

certificatUtilisateur) raises (CertificatIntrouvable);

};

interface AE

{

**private** **attribute** string identifiantAE;

**attribute** short niveau;

Certificat creationDeCertificat(in string nomUtilisateur, in ListUsages usages, in string CP)

raises (CertificatDejaCree) ;

string demandeDeSuspension(in string nomUtilisateur, in Certificat certificatUtilisateur)

raises (CertificatIntrouvable);

string demandeDeRevocation(in string nomUtilisateur, in Certificat certificatUtilisateur)

raises (CertificatIntrouvable);

};

**interface** AC

{

**private** **attribute** string identifiantAC;

**private** **attribute** Certificat certificatAC;

**attribute** short niveau;

**attribute** ListLC LC;

**attribute** ListLC LCR;

Certificat creationDeCertificat(in string nomUtilisateur, in ListUsages usages, in string CP)

raises (CertificatDejaCree);

string demandeDeSuspension(in string nomUtilisateur, in Certificat certificatUtilisateur)

raises (CertificatIntrouvable);

string demandeDeRevocation(in string nomUtilisateur, in Certificat certificatUtilisateur)

raises (CertificatIntrouvable);

string demandeValidationCertificat(in Certificat certificatUtilisateur)

raises (CertificatIntrouvable);

};